MA/CS 375

Fall 2002
Lecture 29
Root Finding

• Given a real valued function $f$ of one variable (say $x$), the idea is to find an $x$ such that:

$$f(x) = 0$$
Root Finding Examples

1) Find real x such that:
\[ x^2 + 4x + 3 = 0 \]

2) Again:
\[ \tanh(x) - 3 = 0 \]

3) Again:
\[ \cos(x) - 2 = 0 \]

Observations on 1, 2 and 3 ?? (two are trick questions)
Requirements For An Algorithmic Approach

- Idea: find a sequence of $x_1, x_2, x_3, x_4, \ldots$ so that for some $N$, $x_N$ is “close” to a root.

- i.e. $|f(x_N)| < \text{tolerance}$

- What do we need?
Requirements For Such a Root-Finding Scheme

• Initial guess: $x_1$

• Relationship between $x_{n+1}$ and $x_n$ and possibly $x_{n-1}, x_{n-2}, x_{n-3}, ...$

• When to stop the successive guesses?
Some Alternative Methods

- Bisection
- Newton’s Method
- Global Newton’s Method
- Avoiding derivatives in Newton’s method.
Bisection

• Recall the following:
  – The intermediate value theorem tells us that if a continuous function is positive at one end of an interval and is negative at the other end of the interval then there is a root somewhere in the interval.
Bisection Algorithm

1) Notice that if $f(a) \times f(b) \leq 0$ then there is a root somewhere between $a$ and $b$

2) Suppose we are lucky enough to be given $a$ and $b$ so that $f(a) \times f(b) \leq 0$

3) Divide the interval into two and test to see which part of the interval contains the root

4) Repeat
Step 1
Step 2

Even though the left hand side could have a root in it we are going to drop it from our search.

The right hand side **must** contain a root!!!. So we are going to focus on it.
Step 3

Which way?

Winning interval

Step 4

Which way?
Bisection Convergence Rate

• Every time we split the interval we reduce the search interval by a factor of two.

• i.e. \[ |a_k - b_k| \leq \left( \frac{|a_0 - b_0|}{2^{k+1}} \right) \]
Individual Exercise

• Code up the *bisection* method

• Starting with the interval $a = -1, b = 1$

• Find the root of $f(x) = \tanh(x-.2)$ to a tolerance of $1e-5$

• Plot iteration number $(k)$ on the horizontal axis and $\text{errRange}(k) = \text{abs}(f(a_k)-f(b_k))$ on the vertical axis
Newton’s Method

• Luxury: Suppose we know the function $f$ and its derivative $f'$ at any point.

• The tangent line defined by:

$$L_c(x) = f(x_c) + (x - x_c)f'(x_c)$$

can be thought of as a linear model of the function of the function at $x_c$
Newton’s Method \textit{cont.}

- The zero of the linear model $L_c$ is given by:
  \[ x_+ = x_c - \left( \frac{f(x_c)}{f'(x_c)} \right) \]

- If $L_c$ is a good approximation to \( f \) over a wide interval then \( x_+ \) should be a good approximation to a root of \( f \)
Newton’s Method cont.

• Repeat the formula to create an algorithm:

\[ x_{n+1} = x_n - \left( \frac{f(x_n)}{f'(x_n)} \right) \]

• If at each step the linear model is a good approximation to \( f \) then \( x_n \) should get closer to a root of \( f \) as \( n \) increases.
First stage:

\[(x_1, f(x_1)) \quad (x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}, 0)\]
Zoom in for second stage

Notice: we are getting closer

(x2, f(x2))

(x3 = x2 - f(x2)/f'(x2), 0)
Convergence of Newton’s Method

• We will show that the rate of convergence is much faster than the bisection method.

• However – as always, there is a catch. The method uses a local linear approximation, which clearly breaks down near a turning point.

• Small $f'(x_n)$ makes the linear model very flat and will send the search far away …
Say we chose an initial $x_1$ near a turning point. Then the linear fit shoots off into the distance!

$(x_1, f(x_1))$

$(x_2=x_1-\frac{f(x_1)}{f'(x_1)},0)$
Newton in Matlab

```matlab
% convergence tolerance
% bail out after 100 iterations

tol = 1e-4;
maxit = 100;

f = 'sin';
fprime = 'cos';

xn = input('First guess to root of fn: ');
fn = feval(f,xn);
dfndx = feval(fprime,xn);

while abs(fn)>tol
    xn = xn-fn/dfndx;
    fn = feval(f,xn);
    dfndx = feval(fprime,xn);
end
```
Team Exercise

- 10 minutes
- Code Newton up
- Test it with some function you know the derivative of.
Newton’s Method Without Knowing the Derivative

• Recall: we can approximate the derivative to a function with:

\[ f'(x) \approx \left( \frac{f(x + \delta) - f(x)}{\delta} \right) \]
tol = 1e-4; % convergence tolerance
maxit = 100; % bail out after 100 iterations
delta = 1e-5; % spacing for derivative

f = 'sin';

xn = input('First guess to root of fn: '); fn = feval(f,xn);
dfndx = (feval(f,xn+delta)-fn)/delta;

while abs(fn)>tol
    xn = xn-fn/dfndx;
    fn = feval(f,xn);
    dfndx = (feval(f,xn+delta)-fn)/delta;
end

'Root', xn

 Modification
Team Exercise

• 10 minutes
• Modify your script to use the approximate derivative (note you will require an extra parameter delta)
• Test it with some function you do not know the derivative of.
Convergence Rate For Newton’s Method

• **Theorem 8** (van Loan p 285)
  
  Suppose $f(x)$ and $f'(x)$ are defined on an interval $I = [x_* - \mu, x_* + \mu]$ where $f(x_*) = 0, \mu > 0$ and positive constants rho and delta exist such that
  
  $|f'(x)| \geq \rho$ for all $x \in I$
  
  $|f'(x) - f'(y)| \leq \delta |x - y|$ for all $x, y \in I$

  $\mu \leq \frac{\rho}{\delta}$

  If $x_c$ is in $I$, then $x_+ = x_c - \frac{f(x_c)}{f'(x_c)}$ is in $I$ and

  $|x_+ - x_*| \leq \frac{\delta}{2\rho} |x_c - x_*|^2 \leq \frac{1}{2} |x_c - x_*|$

  That is $x_+$ is at least half the distance to $x^*$ that $x_c$ was. Also, the rate of convergence is quadratic.
Convergence Rate of Newton’s Method cont

- The proof of this theorem works by using the fundamental theorem of calculus.
- All of the restrictions are important – and can be fairly easily broken by a general function.
- The restrictions amount to:
  1) \( f' \) does not change sign in a neighbourhood of the root \( x^* \)
  2) \( f \) is not too non-linear (Lipschitz condition)
  3) the Newton’s iteration starts close enough to the root \( x^* \) then convergence is guaranteed and the convergence rate is quadratic.
Summary

• We have looked at two ways to find the root of a single valued, single parameter function.

• We considered a robust, but “slow” bisection method and then a “faster” but less robust Newton’s method.

• We discussed the theory of convergence for Newton’s method.